**Learning 02：Trend Description**

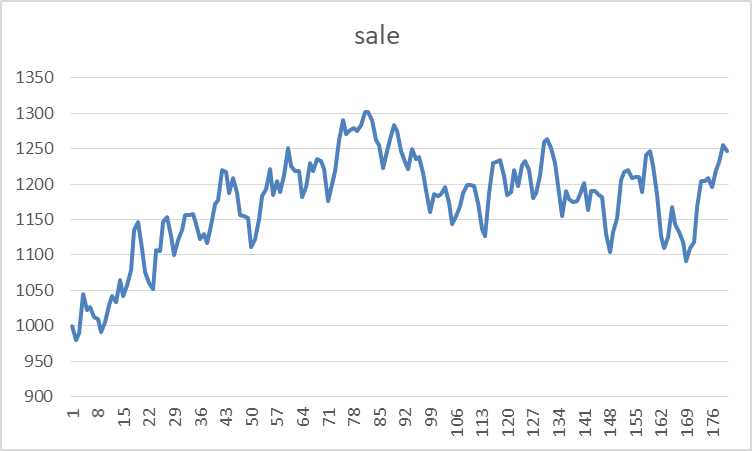
**描述時間趨勢為何重要？**

最初是在Kaggle的〈Top 1% Predict Future Sales, features + LightGBM〉這篇文章中看到對於銷售量趨勢的衡量，覺得相當有趣，因此稍作細部研究。

首先，描述趨勢為何重要？我認為對於趨勢的使用其實深入日常生活與工作需求中。每天早上起床9點看盤，發現台股好像處在一個上下震盪期，已經許久沒有創新高了，此時若處在一個波段高點，根據「過去趨勢」來看可能並不是一個好的進場時間；又或台灣物價持續上漲，曾經麥當勞的蛋捲冰淇淋只要10塊，但現在已經上漲快一倍了，可以預期未來價格也會持續增加。而在工作中，也許今天老闆想要知道某個商品的營收表現，請你做個分析報表，此時最簡單且粗暴的方式就是把這項商品過去歷史銷售資料撈出來，看看它是越賣越好或是越賣越差。

其次，在了解重要性後，下一個問題便是該何描述趨勢？最直觀的方式就是根據時間前後順序，繪製成折線圖，excel在這方面表現得非常好，個人在工作時如果只是想要簡單看的趨勢走向，也會快速將資料貼到excel，並以折線圖呈現，可以很快掌握上升或下降的趨勢，並觀察是否符合當初的預期。

然何在折線圖中，有一個很重要的維度是時間顆粒的粗細，如果是月資料，或許曲線會比較平滑，比較容易說服別人趨勢的走向，但如果資料產生的頻率較高，周資料、日資料或甚至是秒資料，此時曲線會有許多雜音，上下振福會較為劇烈，妨礙圖形的解讀。



除了使用原始資料繪製折線圖外，其實有一些直觀、甚至日常常遇到的方法描繪時間趨勢，這些方法雖然在〈Top 1% Predict Future Sales, features + LightGBM〉一文中是拿來做訓練資料的特徵使用，但直接拿來作區是描述亦是不錯(還可以拿來唬爛別人XD)

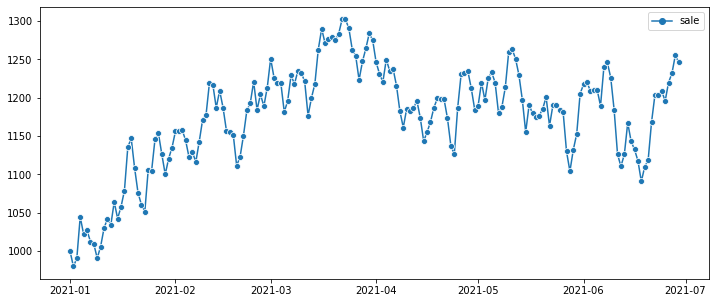
**描述時間趨勢的方法(Python模擬)**

透過Python，可以很快速的模擬銷售數據，我們假設銷售數據是日資料，當t=0時為1000，在模擬資料時，我們預先添加了時間趨勢、週期影響以及噪音。

1. %matplotlib inline
2. **import** numpy as np
3. **import** pandas as pd
4. **import** matplotlib.pyplot as plt
5. **import** seaborn as sns
6. **import** math
8. sale = 1000
9. sales = []
10. **for** day **in** range(180):
11. sales.append(sale)
12. sale = sale + np.random.random() \* 5 + np.sin(math.pi/180 \* day \* (360 / 7)) \* 20 + np.random.normal() \* 20
13. days = pd.date\_range(start ='2021/1/1', periods=180, freq='D')
14. df\_sales = pd.DataFrame(data={'sale': sales}, index=days)

我們可以很快速繪製成折線圖：

1. plt.figure(figsize=(12,5))
2. sns.lineplot(data = df\_sales, markers=True)
3. plt.show()



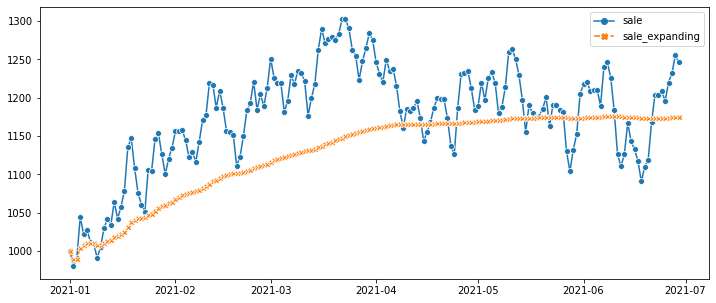
從上圖可以發現，我們會說銷售數字長期而言是呈現增長，然而到了2021-04以後，好像趨勢沒有那麼明顯，同時在我們不知道的情況下，週期與噪音的影響讓折線圖上下起伏較大。

1. **Expanding**

中國大陸翻譯好像是翻常「擴展窗口」，台灣有沒有明確的翻譯我就不清楚了！Expanding的核心概念就是「考量過去的每一期資料，並給予相同的權重」，簡單來說第t期資料的Expanding會計算為第0期至第t期資料的平均值。



1. df\_sales['sale\_expanding'] = df\_sales.sale.expanding().mean()
2. plt.figure(figsize=(12,5))
3. sns.lineplot(data = df\_sales.loc[:, ['sale', 'sale\_expanding']], markers=True)
4. plt.show()



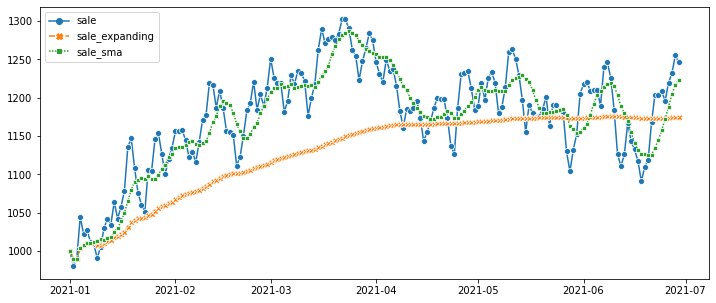
從上圖可以發現，sale\_expanding的曲線平滑很多，可以較清楚了描繪增長趨勢以及在2021-04之後趨於平穩的情況，但它的缺點也相當明顯，由於考量過去所有期數的資料且權重相同，導致曲線會「過於」平滑，而失去了很多短期的趨勢。

1. **Simple Moving Average (簡單移動平均線, SMA)**

SMA的核心概念是將區間內每一個數值相加並且除以總區間數，透過控制區間大小來調整本期會受到過去幾期數據的影響，有效解決Expanding中第0期與第t-1期權重相同的影響，同時透過調整「區間大小(window)」，可以比較短期趨勢與中長期趨勢的差異。



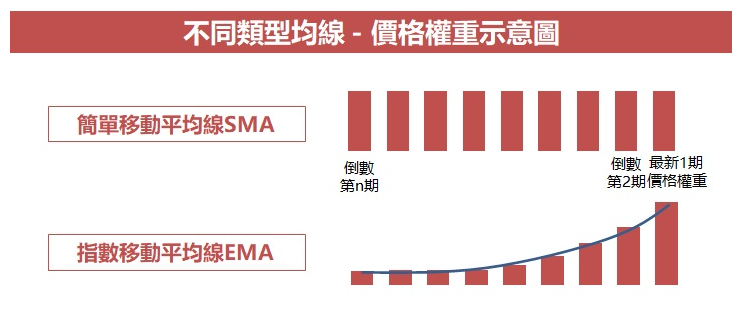
1. df\_sales['sale\_sma'] = df\_sales.sale.rolling(window=7, min\_periods=1).mean()
2. plt.figure(figsize=(12,5))
3. sns.lineplot(data = df\_sales.loc[:, ['sale', 'sale\_expanding', 'sale\_sma']], markers=True)
4. plt.show()

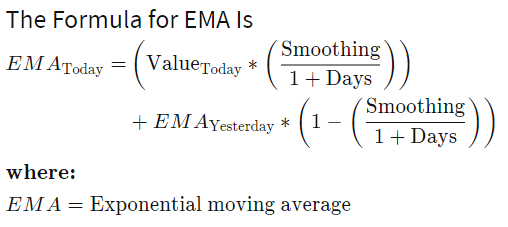


從上圖的sale\_sma曲線(綠線)可以發現，由於我們window=7，因此相較於sale\_expanding(橘線)它捕捉到較為短期的趨勢走勢，但也少掉許多上上下下的噪音。

1. **Exponential Moving Average(指數移動平均線，EMA)**

EMA的核心概念與SMA相近，都是只考慮一定區間(window)的影響，與之相異的點在於SMA對於區間內的數字給予相同權重，而EMA則以指數形式給予權重，與當期越近的權重越大。

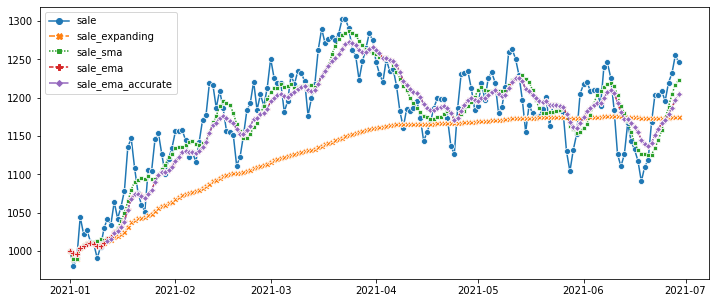




其中一般會以代替，而。

一般而言，如果EMA的span設為4則第0, 1, 2期會是NULL，而第3期的EMA會以SMA代替，然而python的pandas套件中pandas.DataFrame.ewm卻略有不同，它會以第0期的原始值作為第0期的EMA，因此會以理論上的值落有不同，以下我們自訂義一個理論的function作比較。

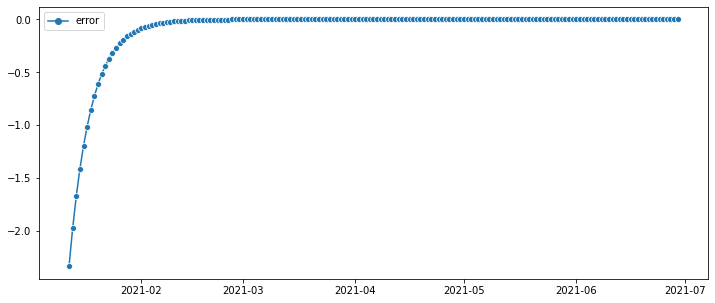
1. **def** ema\_accurate(df\_series, span):
2. df\_ema\_accurate = pd.Series([np.nan] \* (span - 1))
3. df\_ema\_accurate = df\_ema\_accurate.append(pd.Series(df\_series[:span].mean()).append(df\_series[span:]).ewm(span=span, adjust=False).mean())
4. df\_ema\_accurate = df\_ema\_accurate.to\_list()
5. **return** df\_ema\_accurate
6. df\_sales['sale\_ema'] = df\_sales.sale.ewm(span=12, adjust=False).mean()
7. df\_sales['sale\_ema\_accurate'] = ema\_accurate(df\_series=df\_sales['sale'], span=12)
8. plt.figure(figsize=(12,5))
9. sns.lineplot(data = df\_sales.loc[:, ['sale', 'sale\_expanding', 'sale\_sma', 'sale\_ema', 'sale\_ema\_accurate']], markers=True)
10. plt.show()



由於網路上大多觀察短期趨勢時會講span設為12，故同樣以此為主，可以發現或許是因為window=7與span=12的差異，sale\_ema(紅線)與sale\_ema\_accurate(紫線)均較sale\_sma(綠線)更為平滑，然而整體看起來差異其實還好，同時sale\_ema(紅線)與sale\_ema\_accurate(紫線)只有在早期階段略有差異，中後段幾乎完全重合(紅線完全被紫線覆蓋)。

若將兩者的差異繪圖更可以清楚看到它們差異的走勢。

1. cal\_error = pd.DataFrame(data=df\_sales.sale\_ema\_accurate-df\_sales.sale\_ema, columns=['error'])
2. plt.figure(figsize=(12,5))
3. sns.lineplot(data = cal\_error, markers=True)
4. plt.show()



從上圖可以發現，大約經過一個月的時間，兩者的差異已趨近於0。

**結語**

當數據是時間序列的數據時，如何描述時間趨勢就會變成相當重要，當然我們會希望將數據分解成長期時間趨勢、週期趨勢、噪音等，但由於產業知識不足，可能無法在一開始就假設好數據生成的模型，因此初步以折線圖觀察可以提供許多資訊，excel在工作上給予許多便利。

除了純粹原始數據的折線圖外，透過expanding、SMA、EMA可以幫助我們去除許多噪音帶來影響，透過window或span的調整，還可以觀察短期與長期趨勢的差異。